AMENDMENTS TO THE SPECIFICATION

Please replace the paragraph starting on page 4, line 14 with the following:

In one embodiment of the present invention, the request for the password includes computer code that when run on the local computer system requests the password on behalf of the application on the remote computer system. In a variation on this embodiment, the computer code is in the form of a downloadable piece of software or an installed piece of software that runs in the execution environment of the local machine. In a variation on this embodiment, the computer code is in the form of a JAVA™ ~~JAVA~~ applet that runs on a JAVA™ ~~JAVA~~ virtual machine on the local computer system. In a variation on this embodiment, sending the password or the function of the password to the application to the remote computer system involves communicating the password to the JAVA™ ~~JAVA~~ applet, and allowing the JAVA™ ~~JAVA~~ applet to forward the password to the application on the remote computer system.

Please replace the paragraph starting on page 5, line 1 with the following:

In one embodiment of the present invention, the JAVA™ ~~JAVA~~ applet is a signed JAVA™ ~~JAVA~~ applet, and authenticating the request involves authenticating the JAVA™ ~~JAVA~~ applet's certificate chain.

Please replace the paragraph starting on page 8, line 13 with the following:

In the embodiment illustrated in FIG. 1, web browser 130 includes a JAVA VIRTUAL MACHINE™ (JVM) 120 that executes platform-independent instructions in the form of JAVA™ applets comprising JAVA™ ~~JAVA~~ bytecodes. (The terms JAVA, JAVA VIRTUAL MACHINE and JAVA DEVELOPMENT KIT are trademarks of SUN Microsystems, Inc. of Palo Alto, California.)

Please replace the paragraph starting on page 8, line 18 with the following:

B4

JVM 120 executes a number of ~~applets 171-173~~applets 181-183 that originate from applications 161-163, respectively, located on servers 171-173, respectively. Note that ~~applets 171-173~~applets 181-183 are "signed applets." A signed applet includes a digital signature, which is generated and verified through use of a private key-public key pair.

Please replace the paragraph starting on page 9, line 4 with the following:

B5

JVM 120 additionally includes a bytecode verifier 104 that verifies that bytecodes within ~~applets 171-173~~applets 181-183 are properly formed.

Please replace the paragraph starting on page 9, line 6 with the following:

B6

~~Applets 171-173~~Applets 181-183 request passwords on behalf of applications 161-163, respectively, on servers 171-173, respectively. These requests are directed to password lookup mechanism 103, which looks up the requested passwords in password store 102. In one embodiment of the present invention, password ~~store 120~~store 102 is located on client 101. In another embodiment, password store 102 is located within storage device 121 coupled to database server 150. In this embodiment, a user is able to access his or her password store from any location on network 140.

Please replace the paragraph starting on page 9, line 14 with the following:

B7

In one embodiment of the present invention, database server 150 can be accessed by using commands adhering to the lightweight directory access protocol (LDAP). Password ~~store 120~~store 102 can alternatively be stored on a smart card.

Please replace the paragraph starting on page 9, line 21 with the following:

FIG. 2 illustrates the structure of an entry 200 within password store 102 in accordance with an embodiment of the present invention. Entry 200 includes a number of fields, including service name 202, download point 203, user ID 204, password 205, expiry time 206, last login time 207 and last update ~~time 108~~time 208. Service name 202 includes an identifier for the service (or application) that is requesting the password. Download point 203 contains an identifier for the point from which the applet is downloaded to client 101. For example, in FIG. 1 applet 181 is downloaded from server 171. User ID 204 specifies the name of the user that is requesting access to applications 161-163. Password 205 specifies the password associated with user ID 204. Note that user ID 204 is typically sent along with password 205 or a function of the password to an application. Expiry time 206 specifies an expiration time for password 205. Last login time 207 specifies when the user associated with user ID 204 was last logged in to the application specified by service name 202. Last update 208 specifies when password store entry 200 was last updated.

B8

Please replace the paragraph starting on page 10, line 9 with the following:

FIG. 3 is a flow chart illustrating the process of facilitating a single sign on in accordance with an embodiment of the present invention. The system starts at 300, first prompts the user for a single sign on password (step 302), and then uses the single sign on password to open password store 102 (step 304). In one embodiment of the present invention, the process of opening password store 102 includes using the single sign on password to decrypt the password store. Note that the process of opening password store 102 can take place during system initialization. Alternatively, it can take place the first time password store 102 is accessed after system initialization.

B9

Please replace the paragraph starting on page 11, line 1 with the following:

B10

Next, JVM 120 executes ~~applet 161~~applet 181. During this execution, applet 181 requests a password from client 101 on behalf of application 161. This request is received by client 101 (step 310).

Please replace the paragraph starting on page 11, line 14 with the following:

B11

Next, the system uses user ID 204 and service name 202 to look up password 205 in password store 102 (step 312). If password 205 does not exist in password store 102 (step 314), the system adds password 205 to password store 102 (step 316). This may involve prompting the user for password 205, and subsequently adding the password 205 to password store 102.

Please replace the paragraph starting on page 11, line 24 with the following:

B12

Note that the present invention is not limited to using JAVA™ ~~JAVA~~ applets to perform the password lookup. Other embodiments of the present invention use other types of code including ACTIVEX™ and signed ACTIVEX™ ~~ACTIVEX~~ code. (ActiveX is a Trademark of the Microsoft Corporation of Redmond, Washington).

Please replace the paragraph starting on page 12, line 7 with the following:

B13

The system starts starts at 400 and then proceeds by receiving a request to change a password from an application, such as application 161 on server 171 (step 402). In response to this request, the system automatically generates a replacement password (step 404). Since the computer system (and not a human being) generates the replacement password, the replacement password can be quite long and quite random, which makes the password more secure.

Please replace the paragraph starting on page 12, line 19 with the following:

As part of sending the replacement password to ~~application 171~~application 161, the system may additionally send the old password (or a function of the old password) so that ~~application 171~~application 161 can verify that the entity that generated the replacement password was in possession of the old password. The process completes at 410.

B14

Please replace the paragraph starting on page 12, line 25 with the following:

FIG. 5 is a flow chart illustrating the process of opening password store 102 after a time out period has expired in accordance with an embodiment of the present invention. The process starts at 500 and proceeds when a time out period of password store 102 expires (step 502). In response to this expiration, the system again prompts the user for the single sign on password (step 504). The system uses the single sign on password to open the password store (step 506). The process then terminates at 508.

B15

Please replace the Abstract with the following:

One embodiment of the present invention provides a system that facilitates accessing to a plurality of applications that require passwords. When the system receives a request for a password from an application running on a remote computer system, the system first authenticates the request to ensure that it originated from a trusted source. Next, the system uses an identifier for the application to look up the password for the application in a password store, which contains passwords associated with the plurality of applications. If the password exists in the password store, the system sends the password or a function of the password to the application on the remote computer system. Hence, the system creates the illusion that there is a single sign on to a large number of applications,

B16

whereas in reality the system automatically provides different passwords to the applications as they are requested. In one embodiment of the present invention, the request for the password includes computer code that when run on the local computer system requests the password on behalf of the application on the remote computer system. In a variation on this embodiment, the computer code is in the form of a JAVA™ ~~JAVA~~ applet that runs on a JAVA™ ~~JAVA~~ virtual machine on the local computer system. In one embodiment of the present invention, the JAVA™ ~~JAVA~~ applet is a signed JAVA™ ~~JAVA~~ applet, and authenticating the request involves authenticating the JAVA™ ~~JAVA~~ applet's certificate chain.

B16